

Proyecto Final: Analisis Numerico

Maquina Integradora

Pontificia Universidad Javeriana

Analisis Numerico

Presentado por

Alejandro Mayorga

Miguel Paez

Juan Sebastian Barreto

Andrés Mariño

Contenidos

Contenidos	2
Objetivo	3
Marco Teorico	4
Desarrollo	4
Clases	4
Librerías de apoyo	5
Características de GUI	6
Funcionamiento	9
Imágenes de funcionamiento	9
Discusión, inconvenientes, cambios y sugerencias de mejoras	13
Referencias	14

Objetivo

El objetivo de este proyecto es implementar una calculadora especializada que calcula integrales por medio de métodos numéricos, dando opciones al usuario para que personalice la manera en que la integral es calculada, de esa manera la integral es resuelta y también es mostrada una gráfica de la función y el error de truncamiento asociado.

Con este proyecto se busca dar una herramienta a estudiantes o profesionales que necesiten desarrollar cálculos no convencionales que incluyen problemas que no puedan ser resueltos de manera analítica, dando el nivel de precisión que requieran para que los resultados tengan confiabilidad.

Marco Teorico

- Integracion Numerica

La integración por métodos numéricos incluye una variedad de algoritmos y procedimientos para poder calcular el valor numérico de una integral definida, se usa frecuentemente para resolver ecuaciones diferenciales y su utilidad se aprecia al intentar integrar expresiones que no pueden ser resueltas de forma analítica, existen metodos de integracion numérica basados en la interpolación, algoritmos adaptativos y estimación de error.

- Metodo de Riemann

El método de Riemann es un método de integración numérica que consiste en el trazado de una cantidad finita de rectángulos dentro de un área irregular para aproximar el área de la figura, más específicamente calculando el área de cada rectángulo y sumando las áreas.

- Método de Trapecios

Es una variación del método de Riemann que calcula áreas bajo una curva irregular utilizando trapecios con su respectiva fórmula para calcular el área de los mismos.

- Error de truncamiento

Son los errores que resultan de utilizar una aproximación en lugar de un método matemático exacto, la integración numérica al ser un método de aproximación a un resultado debe lidiar con este tipo de errores para que no sean muy grandes, estos errores se evalúan con la serie de Taylor

Desarrollo

- Clases

- GraphApp.Java:

- Clase encargada de graficar las expresiones insertadas en la interfaz de la aplicación.

- GUI_Analisis.Java:

- Clase encargada de implementar la interfaz gráfica del sistema y realizar los cálculos correspondientes a la integración y el error de truncamiento, tiene los siguientes métodos:

- **public Float distancia(float, float):** Metodo encargado de calcular la distancia en valor absoluto entre los límites de integración.

- **public static List<> puntos (float, float, float):** Método encargado de la división del intervalo de integración y asignar la lista de puntos a evaluar en el eje X.
- **Public static List<> puntosY(List<>, String):** Método encargado de asignar cada punto al eje Y en el intervalo de integración evaluando cada punto en la lista de X de acuerdo a la expresión ingresada por el usuario.
- **Public static Float areaRectangular(float, float, List, List):** Método encargado de integrar la expresión tomando las listas de puntos en X y Y, y teniendo en cuenta los límites de integración para formar triángulos y calcular su área (método de Riemann)
- **Public static areaTrapezoidal(float, float, List, List):** Método encargado de integrar la expresión tomando las listas de puntos en X y Y, teniendo en cuenta los límites de integración para formar trapecios y calcular su área (método de trapecios)
- **Public static areaSemicircular(float, float, List, List):** Método encargado de integrar la expresión tomando las listas de puntos en X y Y, teniendo en cuenta los límites de integración para formas semicírculos y calcular su área.
- **Public static List<> particionSin(List<>, float, float):** Metodo encargado de realizar la distribución de puntos en X usando como guía la función $\sin(x)$.
- **Public static list<> xMasUno(List<>, float, float):** Metodo encargado de realizar la distribución de puntos en X usando como guía la función $x + 1$.
- **Public static List<> xALa2(List<>, float, float):** Metodo encargado de realizar la distribución de puntos en X usando como guía la función x^2
- **Public static float errorTruncamientoTrap(float, float, List<>, List<>, String):** Método encargado de calcular el error de truncamiento de la expresión a calcular.

Esta clase cuenta con los siguientes atributos:

- **List <Float> ListX:** Lista de números tipo Float que almacena los puntos en el eje X de la función a evaluar.
- **List <Float> ListY:** Lista de números tipo Float que almacena los puntos en el eje y de la función a evaluar.
- **Float cantidadpuntos:** Cantidad de puntos que serán evaluados para integrar la función, determinado por la resolución del slider en la interfaz gráfica que es modificable por el usuario.
- **Float a:** Límite inferior de integración, determinado por el campo editable en la interfaz gráfica para aceptar valores dados por el usuario.
- **Float b:** Límite superior de integración, determinado por el campo editable en la interfaz gráfica para aceptar valores dados por el usuario.
- **String eq:** Cadena de caracteres editable por el usuario, esté String determina la ecuación a ser evaluada.

- Librerías de apoyo

- Groovy: Librería originalmente de apache e integrable a Java que puede ser usada para convertir un String con una expresión algebraica o una ecuación a una expresión numérica evaluable en Java, con operaciones básicas reconocibles y soportes para parentesis, con la opción de agregar expresiones más avanzadas y su correspondiente traducción a la librería Math de Java. El apoyo de utilizar nuevas funciones se utiliza en el método **PuntosY**, donde se le indica al programa como interpretar expresiones como $\sin()$, $\cos()$, $\tan()$, x^2 , x^3 , \sqrt{x} , etc.

- Características de GUI

Maquina Integradora!

Funcion a integrar:

Grafica de la funcion a integrar:

√ x^2 $\sin()$ $\cos()$ $\tan()$ x^3

Limite inferior (a): 0

Limite superior (b): 10

Area de integracion:

☒ Rectangulos

☐ Trapecios

☐ Semicirculos

Resultado de area:

0

Error de truncamiento:

0

Resolucion:

Integrar!

res: 10

Funcion de particion:

☒ x Constante

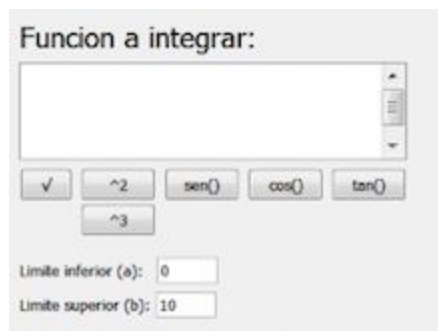
☐ $\sin (x) + 1, x > 0$

☐ $x+1, x > 0$

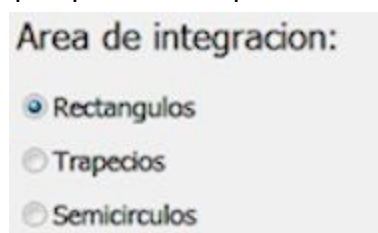
☐ $x^2, x > 0$

Como se puede evidenciar en esta figura, el GUI de nuestro producto tiene varias opciones para integrar expresiones, en esta sección se explica cada componente:

- **Campo de Texto de función y límites de integración:** En el campo grande que se puede observar el usuario ingresa la función que quiere evaluar, es importante tener en cuenta que si desea usar una expresión no regular (función trigonométrica, raíz cuadrada, potencia) se estipula que debe usar los botones debajo del campo de texto para insertar la expresión que desea para que la ecuación sea interpretada correctamente, poniendo entre el paréntesis vacío la incógnita, número o expresión que quiere que sea evaluado, este paréntesis se origina al ingresar la expresión de cada botón, en los campos que se observan más abajo el usuario puede ingresar los valores numéricos que corresponden a los límites de integración.



- **Botones de opción de Área de integración:** Estas opciones implementadas en botones radiales seleccionables dejan escoger al usuario el tipo de área de integración que quieren usar para hacer el cálculo de la integral sobre la función.



- **Área de función de integración:** En este área de la interfaz se muestra la función de integración graficada.

Grafica de la funcion a integrar:

- **Resultado de area y error de truncamiento:** En estos dos campos se despliega el resultado de integración de la expresión ingresada y el error de truncamiento calculado.

Resultado de area:

0

Error de truncamiento:

0

- **Deslizador de resolución:** Usando este deslizador se puede ajustar la resolucion (cantidad de puntos) que el usuario quiere para ajustar la precisión del cálculo.

Resolucion:



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

res: 10

- **Seleccionador de error de partición:** Seleccionando uno de los botones radiales se escoge la función que va a seguir la distribución de puntos en el eje X.

Funcion de particion:

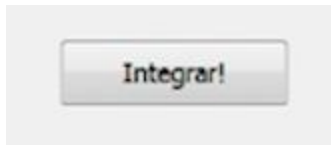
☒ x Constante

☐ $\sin(x) + 1, x > 0$

☐ $x+1, x > 0$

☐ $x^2, x > 0$

- **Botón de integrar:** Con este botón el sistema guarda todas las opciones escogidas y procede a hacer el cálculo correspondiente y mostrar los cálculos hechos en sus campos correspondientes.



Funcionamiento

- Imagenes de funcionamiento

En las siguientes imágenes se muestra el funcionamiento de nuestro software en un par de ejemplos, mostrando la selección de opciones y como se muestran los resultados en la aplicación.

1.

Maquina Integradora!

Funcion a integrar:

Limite inferior (a):

Limite superior (b):

Area de integracion:

☒ Rectangulos

☐ Trapecios

☐ Semicirculos

Resolucion:

2830

Funcion de particion:

☐ x Constante

☐ $\sin(x) + 1, x > 0$

☒ $x+1, x > 0$

☐ $x^2, x > 0$

Grafica de la funcion a integrar:

Resultado de area:

Error de truncamiento:

Se integrara con una precision de 7 decimales

4.

Maquina Integradora!

Funcion a integrar:

Limite inferior (a):

Limite superior (b):

Area de integracion:

☒ Rectangulos

☐ Trapecios

☐ Semicirculos

Resolucion:

2830

Funcion de particion:

☐ x Constante

☐ $\sin(x) + 1, x > 0$

☒ $x+1, x > 0$

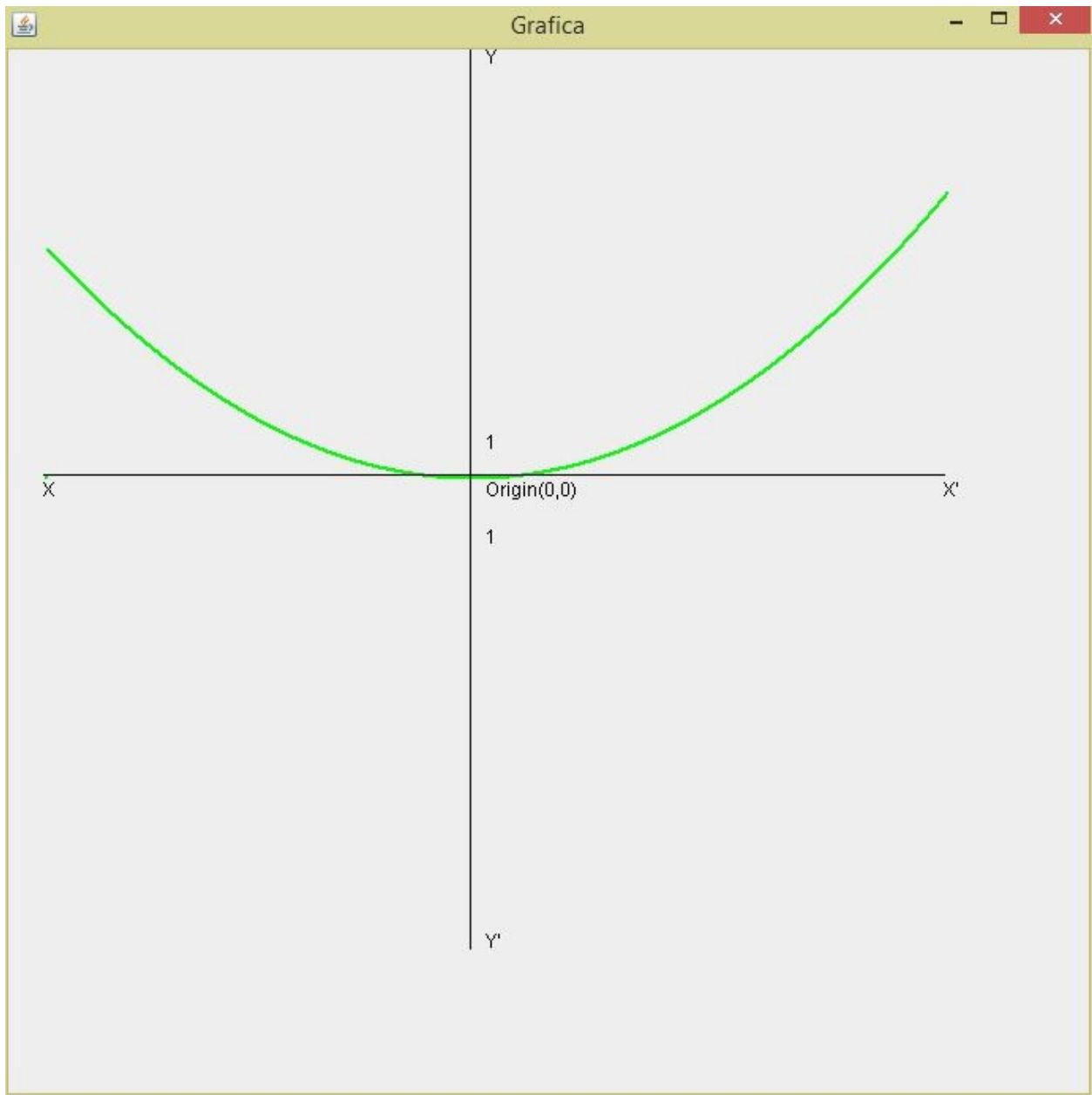
☐ $x^2, x > 0$

Grafica de la funcion a integrar:

Resultado de area:

Error de truncamiento:

5.



- Discusión, inconvenientes, cambios y sugerencias de mejoras

En el desarrollo de nuestro proyecto hubieron dificultades grandes respecto a el cálculo de error de truncamiento y de gráficas debido el lenguaje escogido para implementar el programa y el manejo del tiempo, las dificultades en cuanto a concepto o cosas técnicas no se presentaron demasiado ya que el proyecto es muy específico en cuanto a lo que quiere hacer.

Al escoger el lenguaje JavaScript se tuvo que invertir una cantidad importante de tiempo en lograr que el programa pudiera interpretar una cadena de caracteres como una expresión matemática siguiendo las reglas de operadores e identificando funciones complejas, para esto se recurrió a una librería externa que ayudó a interpretar este tipo de cadenas de caracteres y extendiendo su entendimiento a funciones que creemos son relevantes al contexto de nuestra aplicación. La parte gráfica quizá fue la que más dificultad representó en el proyecto debido a la escasez de recursos para poder implementar este tipo de funcionalidad.

Referencias

1. 4, e., 4, e., & perfil, V. (2008). *TIPOS DE ERRORES*. *Meto2numericos.blogspot.com.co*. Retrieved 22 May 2018, from <http://meto2numericos.blogspot.com.co/2008/02/tipos-de-errores.html>
2. (2018). *Ing.unne.edu.ar*. Retrieved 22 May 2018, from <http://ing.unne.edu.ar/computacion/pub/informatica/IN.pdf>
3. *Revista Digital Matematica, Educacion e Internet*. (2018). *Tecdigital.tec.ac.cr*. Retrieved 22 May 2018, from <https://tecdigital.tec.ac.cr/revistamatematica/>
4. (2018). *Dspace.espol.edu.ec*. Retrieved 22 May 2018, from <https://www.dspace.espol.edu.ec/bitstream/123456789/20886/1/ANALISIS%20NUMERICO%20BASICO.pdf>